JavaScript Quick Reference

Introduction to JavaScript

JavaScript is a versatile programming language used to create dynamic and interactive content on websites. It allows developers to build web applications, games, and much more.

Variables

Variables store data that can be used and modified later.

Declaration:
let name = "Gabby"; // Changeable variable
const age = 25; // Cannot be reassigned
var city = "Toronto"; // Older syntax (not recommended)

Data Types

Common JavaScript data types include:

- String: Text (e.g., "Hello")

- Number: Numbers (e.g., 42, 3.14)

- Boolean: True/false (e.g., true, false)

- Array: List of items (e.g., [1, 2, 3])

- Object: Key-value pairs (e.g., {name: "Gabby", age: 25})

Example:
let isLearning = true;
let colors = ["red", "blue", "green"];
let user = { name: "Gabby", age: 25 };

Operators

JavaScript operators include:

- Arithmetic: +, -, *, /, %

- Comparison: ==, !=, ===, !==, >, <, >=, <=

- Logical: &&, ||, !

Example:
let result = (10 > 5) && (3 < 8); // true

Functions

Functions are reusable blocks of code.

Declaration:

```
function greet(name) {
  return `Hello, ${name}!`;
}

console.log(greet("Gabby")); // Hello, Gabby!
```

Arrow Functions:

```
const add = (a, b) => a + b;
console.log(add(5, 3)); // 8
```

Conditional Statements

Control the flow of code based on conditions.

Example:

```
let score = 85;

if (score >= 90) {
  console.log("A grade");
} else if (score >= 80) {
  console.log("B grade");
} else {
  console.log("Try again");
}
```

Loops

Loops allow repeated execution of code.

Example:

```
for (let i = 0; i < 5; i++) {
  console.log(i); // Prints 0 to 4
}

let count = 0;
while (count < 3) {
  console.log(count);
  count++;
}
```

DOM Manipulation

JavaScript can dynamically change content on a webpage.

Example:

```
document.getElementById("myButton").addEventListener("click", () => {
  document.getElementById("myText").textContent = "Button clicked!";
});
```

Events

Events are user interactions with a webpage.

Example:

```
const button = document.querySelector("button");

button.addEventListener("mouseover", () => {
  console.log("Mouse is over the button!");
});
```

Array Methods

JavaScript provides methods to manipulate arrays, such as:

- push(): Add to the end of an array.

- pop(): Remove the last element.

- map(): Apply a function to each item.

- filter(): Create a new array with elements that pass a condition.

Example:

```
let numbers = [1, 2, 3, 4, 5];
let evenNumbers = numbers.filter(num => num % 2 === 0);
console.log(evenNumbers); // [2, 4]
```

Objects

Objects store related data and functions.

Example:

```
let car = {
  brand: "Toyota",
  model: "Corolla",
  year: 2020,
  start: function() {
    console.log("Car started!");
  }
```

```
};

console.log(car.brand); // Toyota
car.start(); // Car started!
```

Fetch API

The Fetch API is used to make HTTP requests.

Example:

```
fetch("https://api.example.com/data")
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error("Error:", error));
```

Practice Exercise

1. Create a function that calculates the area of a rectangle.

2. Create an array of numbers and use map() to double each number.

3. Create a button that changes its text when clicked.

Additional Resources

- W3Schools JavaScript Tutorial: https://www.w3schools.com/js/

- MDN Web Docs on JavaScript: https://developer.mozilla.org/en-US/docs/Web/JavaScript

Final Words

JavaScript is a powerful language for building interactive web applications. Practice these basics to build a strong foundation and explore advanced features as you grow.